CPS 130 Homework 5 - Solutions

1. Give asymptotic upper and lower bounds for the following recurrences. Assume $T(n)$ is constant for $n = 1$. Make your bounds as tight as possible, and justify your answers.

   (a) $T(n) = 2T(n/4) + \sqrt{n}$
   Solution: By the Master Theorem:

   - $a = 2$, $b = 4$, $c = 1/2$
   - $a = 2 = \sqrt{4} = b^c$
   - $T(n) = \Theta(\sqrt{n} \log_4 n)$

   (b) $T(n) = 7T(n/2) + n^3$
   Solution: By the Master Theorem:

   - $a = 7$, $b = 2$, $c = 3$
   - $a = 7 < 8 = b^c$
   - $T(n) = \Theta(n^3)$

   (c) $T(n) = 7T(n/2) + n^2$
   Solution: By the Master Theorem:

   - $a = 7$, $b = 2$, $c = 2$
   - $a = 7 > 4 = b^c$
   - $T(n) = \Theta(n^{\lg 7})$

   (d) $T(n) = 5T(n/5) + n/\log n$
   Solution: The Master Theorem does not apply. For simplicity we may assume $\log n = \log_5 n$ and solve by iteration:

$$\begin{aligned} T(n) &= n/\log_5 n + 5T(n/5) \\ &= n/\log_5 n + n/\log_5 (n-1) + 5^2 T(n/5^2) \\ &= \ldots \\ &= n/\log_5 n + n/\log_5 (n-1) + \ldots + n/\log_5 (n-k) + 5^{k+1} T(n/5^{k+1}). \end{aligned}$$

   Note that $n/5^{k+1} = 1$ when $k = \log_5 n - 1$. Then,

$$\begin{aligned} T(n) &= \sum_{k=0}^{\log_5 n - 1} \frac{n}{\log_5 (n-k)} + \Theta(1) \\ &= \sum_{k=1}^{\log_5 n} \frac{n}{k} + \Theta(1) \\ &= n \sum_{k=1}^{\log_5 n} \frac{1}{k} + \Theta(1) \\ &= \Theta(n \ln \log_5 n) \end{aligned}$$

2. (CLRS 7.1-2) What value of q does PARTITION return when all elements in the array $A[p..r]$ have the same value? Modify PARTITION so that $q = (p+r)/2$ when all elements in the array $A[p..r]$ have the same value.

   Solution: The original partition element will return its index in the array which will be q. This element as defined in PARTITION will be the last index of the array sent into the function, i.e. $q = r$. To modify PARTITION, add a check for equality of n at the beginning of the code. If all of the values are equal, then return the middle index $q = (p+r)/2$. This will take $O(n)$ time and will not increase the running time of the algorithm.

3. (CLRS 7.2-3) Show that the running time of QUICKSORT is $\Theta(n^2)$ when the array A contains distinct elements and is sorted in decreasing order.

   Solution: On the first iteration of PARTITION the pivot element is chosen as the first element of A. Index i is incremented once and j is decremented until it reaches the pivot, i.e. the entire length of A. PARTITION returns to QUICKSORT the first element of A, which recursively sorts one subarray of size 1 and one of size $n-1$. This process is repeated for the subarray of size $n-1$. The running time of the entire computation is then given by the recurrence:

$$
\begin{aligned}
T(n) &= \begin{cases} \Theta(1) & n \leq 2 \\ T(n-1) + \Theta(n) & \text{otherwise} \end{cases} \\
&= \Theta(n^2).
\end{aligned}
$$